

## Toward accurate Amazigh part-of-speech tagging

Rkia Bani<sup>1</sup>, Samir Amri<sup>2</sup>, Lahbib Zenkour<sup>1</sup>, Zouhair Guennoun<sup>1</sup>

<sup>1</sup>Smart Communication Research Team (SCRT), Mohammadia School of Engineering (EMI), Mohammed V University, Rabat, Morocco

<sup>2</sup>Department of Computer Science and Mathematics, ENSAM School of Engineering, Moulay Ismail University, Meknes, Morocco

### Article Info

#### Article history:

Received Jan 17, 2023

Revised Jun 1, 2023

Accepted Jun 3, 2023

#### Keywords:

Amazigh language

Machine learning

Natural language processing

Part-of-speech tagging

Recurrent neural network

### ABSTRACT

Part-of-speech (POS) tagging is the process of assigning to each word in a text its corresponding grammatical information POS. It is an important pre-processing step in other natural language processing (NLP) tasks, so the objective of finding the most accurate one. The previous approaches were based on traditional machine learning algorithms, later with the development of deep learning, more POS taggers were adopted. If the accuracy of POS tagging reaches 97%, even with the traditional machine learning, for high resourced language like English, French, it's far the case in low resource language like Amazigh. The most used approaches are traditional machine learning, and the results are far from those for rich language. In this paper, we present a new POS tagger based on bidirectional long short-term memory for Amazigh language and the experiments that have been done on real dataset shows that it outperforms the existing machine learning methods.

*This is an open access article under the [CC BY-SA](#) license.*



### Corresponding Author:

Rkia Bani

Smart Communication Research Team, Mohammadia School of Engineering (EMI), Mohammed V University  
Rabat, Morocco

Email: rkianbani@research.emi.ac.ma

## 1. INTRODUCTION

Part-of-speech (POS) tagging is called the process of marking up each word (token) in a text (corpus) with a convenient POS depending on its definition or its context. Simple tagging consists of assigning nouns, verbs, adjectives to words, sophisticated tagging specifies more the tag with, for example, its gender, number POS tagging is an important pre-processing task for other natural language processing (NLP) tasks. It is primary for other applications used in the NLP such as information retrieval, questions-answering systems, information extraction, summarization, machine translation. There are two main approaches for pos tagging: The rule-based approach, where rules are handwritten to be implemented in algorithms. Hence, it's time and resource consuming. On the other hand, there is a machine learning approach based on stochasticity and probabilities. But with the development of the machine's capacity and the explosion of data, deep neural networks have become the popular approach for resolving the most NLP tasks. Although the traditional machine learning methods have delivered good results for high resourced language, low resourced languages haven't yet reached these results specifically Amazigh language.

The Amazigh language, more common as Tamazight, is belonging to the Afro-Asiatic (Hamito-Semitic) family languages [1], [2]. It spreads in the Northern region of Africa that expands from the Niger in the Sahara to the Mediterranean Sea to the Canary Isles. In the aim of providing an adequate standard writing system, the Tifinagh- Institut de Recherche et Coordination Acoustique/Musique (IRCAM) graphical system has been elaborated to represent the Moroccan Amazigh language in the best way. The Tifinaghe-IRCAM system includes,

- 27 consonant covering: the dentals (ⵜ, ⵍ, ⵎ, ⵏ, ⵐ, ⵑ, ⵒ, ⵓ, ⵔ), the labials (ⵙ, ⵚ, ⵛ), the velar (ⵖ, ⵗ), the alveolars (ⵝ, ⵞ, ⵟ, ⵠ), the labiovelars (ⵙ, ⵚ), the palatals (ⵛ, ⵜ), the pharyngeals (ⵡ, ⵢ), the uvular (ⵣ, ⵤ) and the laryngeal (⵨).
- Two semi-consonants: ⵝ and ⵞ.
- 4 vowels: ⵏ, ⵓ, ⵔ, ⵉ known as full vowels, and one neutral vowel ⵉ that is considered a genuineness of the Amazigh phonology.

The most syntactic classes of the Amazigh language are noun, noun is a grammatical class obtained from the combination of a root and a pattern. It might be in a simple form (ⵓⴰⵔⵉⵎ 'aghrum' the hand), compound form (ⵉⵎⵓⵔⵉⵎ 'buEari' the forest keeper) or derived one (ⵓⴰⵎⵎⴰⵏ 'aslmad' the teacher). The nouns could be masculine or feminine, they could be singular or plural and they could be in free case or construct case. In the other hand verb, in Amazigh could be basic or derived. In the case of the basic form, verb is composed of a root and a radical. In the case of the derived form, verb has a basic form attached to one of the prefixes morphemes: ⵜⵜ 'tt' that marks the passive form, ⵓⵔ 's' / ⵓⵔⵔ 'ss' which indicates the factitive form, and ⵎⵎ 'm' / ⵎⵎⵎ 'mm' that designates the reciprocal form. These two types of verbs are both conjugated in four aspects: perfect, negative perfect, aorist and imperfective. Finally, there is particle which is a function word that can't designate noun or verb. It includes conjunctions, pronouns, aspectual, prepositions, orientation, negative particles, subordinators, and adverbs. Usually, particles are uninflected words except for the demonstrative and possessive pronouns (ⵡⵓ 'wa' this (mas.) ⵡⵉⵏ 'win' these (mas.)).

Although high resourced or well-studied language have different POS tagging systems whether in classical machine learning or in deep learning, it is always a challenging task for Amazigh as it is a very low resource language. The reason for this is due to the lack of linguistic resources such as morphological analyzer and annotated corpus, the first corpus elaborated counts just 20k token, then the second [3] with 60k token. All these challenges motivate researchers to bring in more solutions for Amazigh text analyzing. Recently, deep learning offered more solutions for different NLP especially bidirectional recurrent network. In this context, we propose for the first time in Amazigh processing text a POS tagger based on bidirectional long short term memory.

Since the importance of POS tagging in NLP, NLP researchers have been interested in it for the past decades. Early approaches were based on probabilities such as hidden Markov models (HMM) [4] and conditional random fields (CRF) [5]. Or based on statistics such as support vector machine (SVM) [6]. Recently, with the expansion of data in the net, the deep learning models recognize an important development starting from recurrent networks to bidirectional ones that we will present thoroughly in the next section.

The first research on Amazigh POS tagging was done by [7] using the first annotated corpus of just 20k tokens, it was based on traditional machine learning, such as CRF and SVM with an accuracy of 88.66% and 88.26 % respectively. Later, with the elaboration of another corpus of 60k tokens [3], the CRF and SVM algorithms are tested again in [8] giving an accuracy of 89% and 88% respectively. In addition, tree tagger as an independent language tagger has been tested in the Amazigh language in [8] with an accuracy of 89%. As we see, the accuracy of tagging the Amazigh language is still low compared to the well-known language. In the Germain language tree tagger reaches 96% [9], as for French reaching 97% [10]. Certainly, the availability of the annotated corpus is the reason behind this large difference in accuracy between the Amazigh language as a low resource and the other rich language. As we are talking about 300k tokens range compared to 60k tokens. As traditional machine learning algorithms have already been tested in the Amazigh language, we are wondering if deep learning could enhance the tagging accuracy for this low resource language specifically the bidirectional long-short-term-memory (Bi-LSTM).

Recently, Bi-LSTM [11], [12] networks have been the center of interest in multiple NLP tasks, starting from sentiment analysis [13] and semantic role labeling [14] to dependency parsing [15], [16]. In syntactic chunking, authors in [17] are the first to use a BI-LSTM in addition to the CRF layer to sequence tagging with state of art results. In the matter of POS tagging, in [18], they used BI-LSTM to test POS tagging on multiple languages and obtained the existing state-of-the-art results. On the other hand [19] developed a unified tagging Bi-LSTM model for chunking, POS tagging and named-entity-recognition. However, in [20] they used the Bi-LSTM network not only in rich languages but also in some languages with a corpus of just 80k tokens, and the results were promising. Bidirectional LSTMs read a sequence of inputs on both senses forward and backward before passing on to the next layer. For more information see [21], [22]. The main contribution of this work includes,

- Present the challenges of POS tagging low resource language such as Amazigh and discuss the state-of-art approach already used to solve this task.
- Propose a new model for POS tagging Amazigh which is Bi-LSTM and verify its performance on the Amazigh dataset.

- Compare the performance of our tagger with the existing taggers and demonstrate that our tagger offers state-of-the-art results.

## 2. METHOD

In this section, we present the architecture of the model that we used in this paper. It's based on Bidirectional LSTM with an embedding layer. And to choose the best parameters, we realized several experiments on the size of the cell as well as on the embedding size.

### 2.1. BI-LSTM architecture

A recurrent neural network (RNN) is a type of neural network that enables the use of previous predictions as input, thanks to the hidden state, hence modeling the contextual information dynamically. Beginning with an input sequence  $x_1, x_2, x_n$ , a standard RNN estimates the output vector  $y_t$  of each word  $x_t$  by calculating at each step the (1) and (2).

$$h_t = A(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

$$y_t = W_{hy}h_t + b_y \quad (2)$$

Where  $W$  represents the weight matrix connecting two layers (like  $W_{hy}$  is the weights between the output and the hidden layer),  $h_t$  is the vector of hidden states,  $b$  represents the bias vector (e.g.:  $b_y$  is the bias vector of the output layer) and  $A$  is the activation function of the hidden layer. We point out that  $h_t$  contains information generated from the output of the hidden state  $h_{t-1}$ , so that RNN can exploit all input history. Nevertheless, the number of the input history that can be used is practically limited, since the impact of certain inputs may decline or explode exponentially through the hidden states, this phenomenon is called the vanishing gradient problem [23]. To remediate this problem the long short-term memory (LSTM) architecture has been established [12].

An LSTM network, presented in Figure 1, enables the standard RNN to keep a memory of inputs for a long time. LSTM network has three gates as shown in Figure 1: an input gate ( $i_t$ ), a forget gate ( $f_t$ ) which decides to keep or not a memory of precedent cell and an output gate ( $o_t$ ), all gathered in a memory block as shown in Figure 1. To compute the output of the LSTM hidden layer  $h_t$  starting from an input  $x_t$ , we solve the (3)-(7) [24].

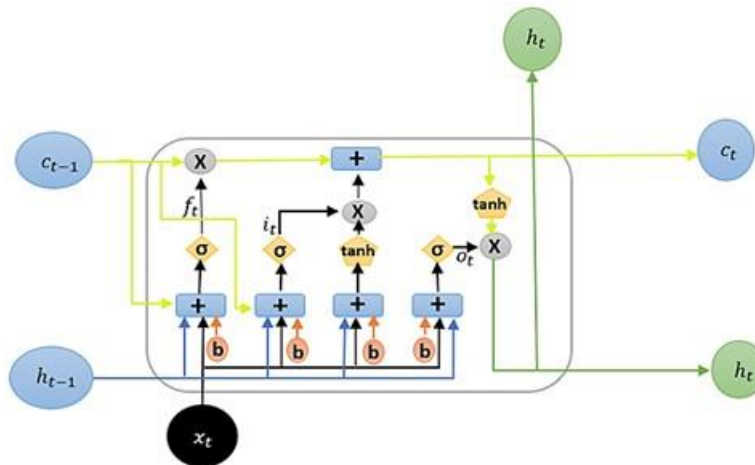


Figure 1. Schematic representation of a typical LSTM network

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (4)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (5)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \quad (6)$$

$$h_t = o_t \tanh(c_t) \quad (7)$$

Note that  $\sigma$  represents the logistic sigmoid function, and i, o, f and c are correspondingly the input gate, the output gate, the forget gate and the cell of activation for vectors. The arrows in Figure 1 represent the weights matrices. Thanks to these multiple gates, the LSTM network can remember periods of time, with the size depending on the chosen architecture, and so it offers a solution of the problem known as vanishing gradient. To extend the knowledge about the more the architecture of LSTM, see [25].

Not only Bi-LSTM networks [11] solved the problem of vanishing gradient but also provided both the preceding and succeeding context. Which was not possible in conventional RNN. This advantage of Bi-LSTM is very helpful on a task like POS tagging where the whole sentence is given. As illustrated in Figure 2, Bi-LSTM computes first the forward hidden sequence  $\vec{h}_t$  and the backward hidden sequence  $\overleftarrow{h}_t$ , then combines  $\vec{h}_t$  and  $\overleftarrow{h}_t$  to generate the output  $y_t$ . We can describe this operation with these equations,

$$\vec{h}_t = H(w_{x\vec{h}}x_t + w_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}}) \quad (8)$$

$$\overleftarrow{h}_t = H(w_{x\overleftarrow{h}}x_t + w_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}) \quad (9)$$

$$y_t = w_{\vec{h}y}\vec{h}_t + w_{\overleftarrow{h}y}\overleftarrow{h}_t \quad (10)$$

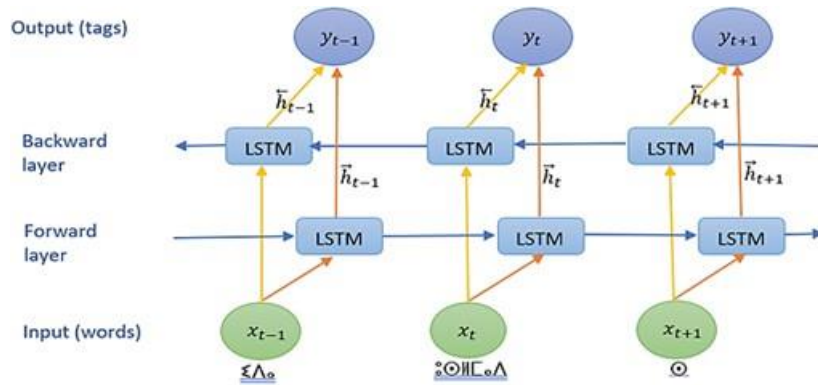


Figure 2. Schematic representation of Bi-LSTM architecture

## 2.2. Our proposed model

The proposed model shown in Figure 3, includes three layers, a word embedding [26] layer, a bidirectional layer with LSTM cells, and finally a SoftMax layer. Starting with a sentence of  $n$  words having tags:  $y_1, y_2, \dots, y_n$ , the Bi-LSTM tagger is first trained to predict the tags. The sentence is transformed to  $[w_1, w_2, \dots, w_n]$  where  $w_i$  is the code number given to the word at index  $i$  according to the vocabulary. This sequence is first entered into the word embedding layer and then fed to the bidirectional layer to finally SoftMax the results to fulfill the predictions. The input of the word embedding layer is designed in a way that if even the dataset changes, the vocabulary received by this layer will be taken into consideration. As for the word embedding output, the number of LSTM cells, the parameters are chosen in such a way that the model performs better.

### 2.2.1. Dataset and tag set

The dataset used in this experiment is the one elaborated by [3], it is a collection of Amazigh texts from various resources. It is a CSV file of order 60k tokens including two columns of word and a tag, Table 1 shows the statistics of this used dataset. To use this corpus well, a preprocessing step was taken to allow us to use it in machine learning models. The first step is to delimit the sentence and to assign each word an ID, the second step is to transform the corpus into a list of sentences in this format: [(word1, tag1), (word2, tag2), (wordN, tagN)] to be ready for the next processing steps such as encoding words into indices and tags to one hot encoding.

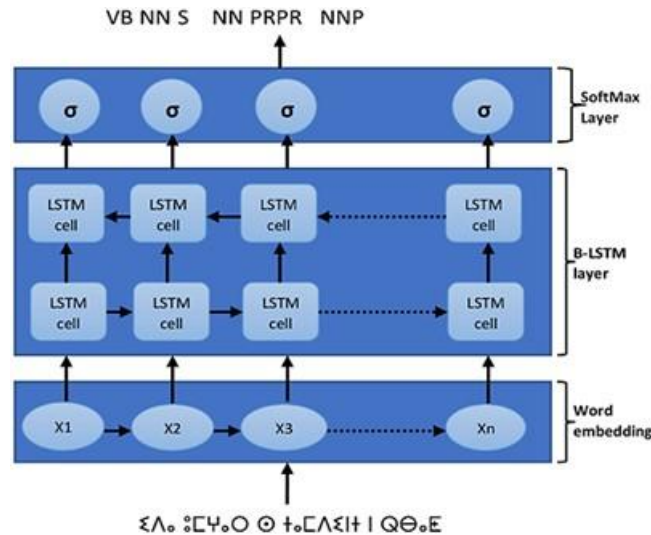


Figure 3. The Bi-LSTM model proposed for Amazigh POS tagging

Table 1. Amazigh dataset statistics

Dataset	Size	Type	Number of sentences	Number of tokens
Monolingual corpus	700 MB	Utf-8	3231	60000

The tag set is a collection of POS tags (labels) to indicate the pos tag or some other grammatical categories of each token in text. The Amazigh language has its specific tag-set represented in Table 2; it is the same used in previous Amazigh NLP works like [3]. This tag set is like other languages with the variance of having multiple particles.

Table 2. The amazigh tag-set

No	TAG	Designation	No	TAG	Designation
1	NN	Common noun	15	PROR	Particle, orientation
2	NNK	Kinship noun	16	PRPR	Particle, preverbal
3	NNP	Proper noun	17	PROT	Particle, other
4	VB	Verb, base form	18	PDEM	Demonstrative pronoun
5	VBP	Verb, participle	19	PP	Personal pronoun
6	ADJ	Adjective	20	PPOS	Possessive pronoun
7	ADV	Adverb	21	INT	Interrogative
8	C	Conjunction	22	REL	Relative
9	DT	Determiner	23	S	Preposition
10	FOC	Focalizer	24	FW	Foreign word
11	IN	Interjection	25	NUM	Numerical
12	NEG	Particle, negative	26	DATE	Date
13	VOC	Vocative	27	ROT	Residual, other
14	PRED	Particle, predicate	28	PUNC	Punctuation

### 2.2.2. Experimental environment

The proposed model is developed using a layer for word embedding, and a bidirectional LSTM layer then a SoftMax layer. For the implementation purpose, we used the TensorFlow library with Keras application programming interface (API). Model optimization is a primary task for the implementation of a deep learning model, it's done by choosing some hidden parameters that influence the model's behavior. Those hyperparameters are embedding size, direction, number of LSTM cells, number of epochs, and batch size.

### 2.2.3. Parameters configuration

In this experiment, the size of the input layer is fixed to 100 and the output layer is 28 as we have 28 tags. To choose the embedding size, we run different experiments changing its size, the results are shown in Figure 4(a). The accuracy changes are very close, we have chosen the size 110 as it's the best performing. On the other hand, for the number of hidden layers, we evaluate the performance of the Bi-LSTM model on

different hidden layer sizes. According to Figure 4(b), the best performance is at size 64. To summarize, the different hyperparameters and their values are shown in Table 3.

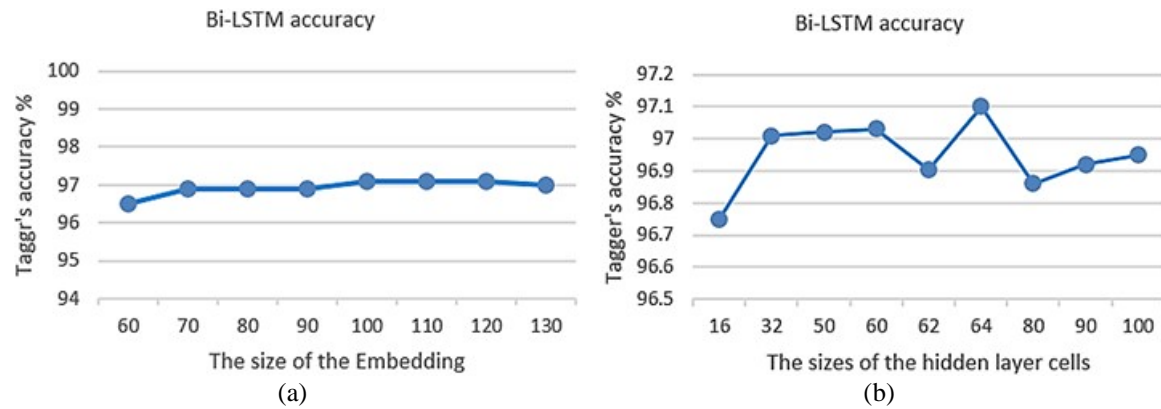


Figure 4. Performance of Bi-LSTM with; (a) different embedding size and (b) different hidden layer size

Table 3. List of the hyperparameters values used in the proposed model

Dataset	Sentences	Word embedding	Hidden layer	Epochs	Batch size	activation
Training	2334	110	64	50	128	SoftMax
Validation	412					
Test	485					

### 3. RESULTS AND DISCUSSION

In this section, we present the result that we get using the detailed specifications in Table 3. We must emphasize that the Amazigh language hasn't yet a trained word embedding. So, in this experiment, under the Keras embedding layer, we will test the Bi-LSTM model without training embedding by randomly initialized word embedding, then we will train the word embedding layer by enabling it in the model.

Figure 5 represents the accuracy and loss function of our Bi-LSTM model with training embedding. As we can see in Figure 5(a) the accuracy reaches 97.10% and in Figure 5(b) the loss is less than 11%. For the nontrained word embedding model, it requires setting up the number of epochs to have significant results, so we test it with 200 epochs and the results were promising too, as in Figure 6(a) the accuracy reaches 94.8% and in Figure 6(b) loss is 17% as shown in Figure 6. To summarize the different experiments realized with this model such as the number of training parameters, and the accuracy obtained, we dress Table 4.

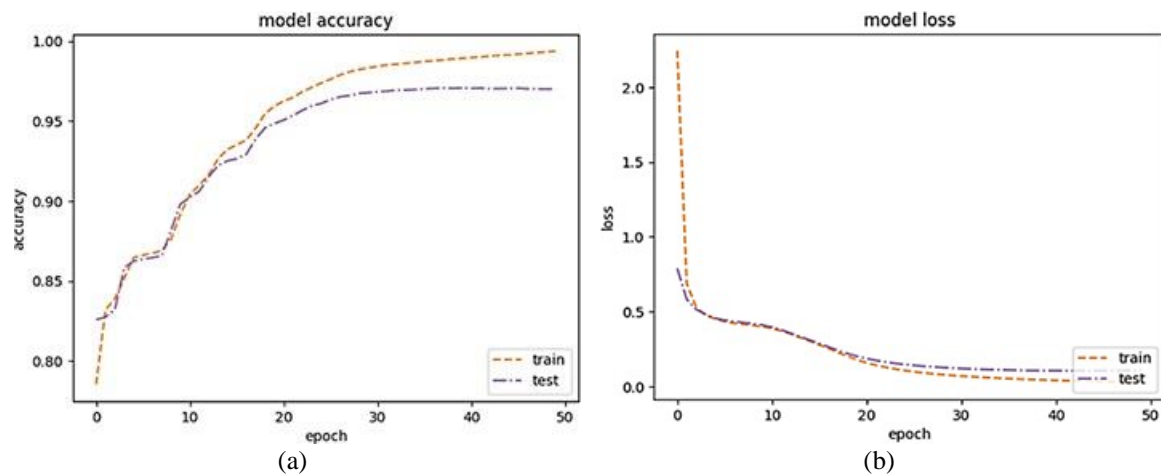


Figure 5. performances of Bi-LSTM model with training word embedding for; (a) Accuracy for and (b) Loss



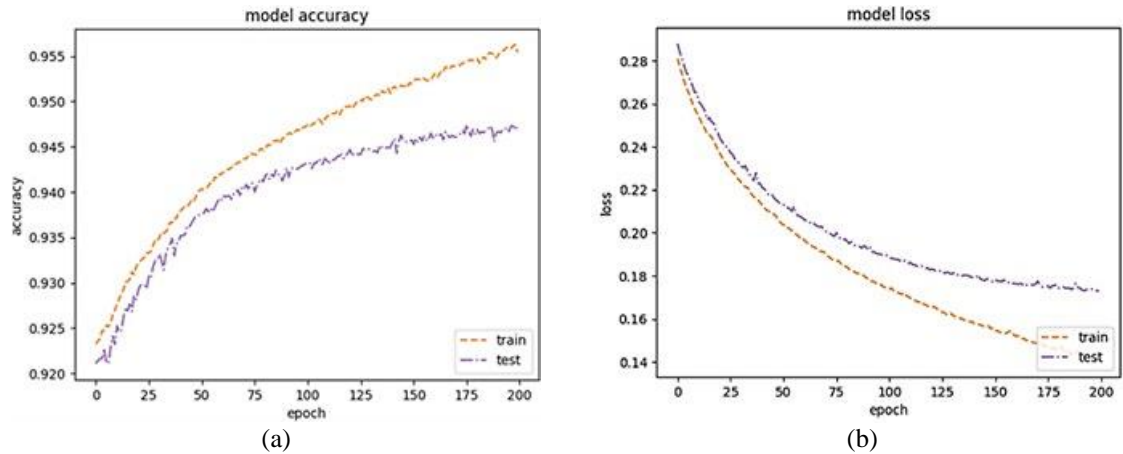


Figure 6. performances of Bi-LSTM model without training word embedding for; (a) Accuracy for and (b) Loss

Table 4. Summary of the experiment specifications and results

Model	Trained parameters	Epochs	Accuracy	Loss
Bi-LSTM with training embedding	1,338,192	50	97.10%	10%
Bi-LSTM without training embedding	93,212	200	94.8%	17%

For this experiment, we have chosen to compare our Bi-LSTM tagger with those Amazigh taggers cited in the section of related works as the baseline. The first SVM and CRF approaches use a 20k tokens corpus. The second CRF, SVM approaches using a 60k tokens corpus with another tree tagger approach. Table 5 shows the different approaches used and their results compared to our model.

Table 5. Comparison of our Bi-LSTM Tagger to the baseline

Corpus	Tagger	Accuracy (%)
20k tokens	CRF [7]	88.66
	SVM [7]	88.26
60k tokens	CRF [8]	89
	SVM [8]	88
	Treetagger [8]	89
Our approach	Our Bi-LSTM without training embedding	94.8
	Our Bi-LSTM with training embedding	97.10

This comparison proves that our proposed model for tagging Amazigh language, specifically, Bi-LSTM outperforms the existing tagger for this language. Even without training the embedding, our model outperforms the traditional machine learning used in Amazigh language tagging. Figure 7 is a graphic representation of those results, and as we can see the outperformance of our model even with the size of the dataset that can be considered small regarding other languages.

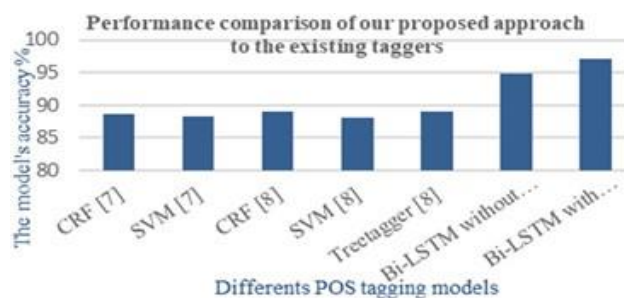


Figure 7. Graphic representation of the comparison of the performance of baseline and our proposed model

#### 4. CONCLUSION

While POS tagging in rich language reaches 97% with traditional machine learning models, Amazigh as a low resource language remains searching for a good pos tagger. In this paper, we presented a new tagger based on Bi-LSTM model that performs the state-of-art for the Amazigh language even without pre-trained word embedding. In future work, we will keep enriching the Amazigh language by creating new datasets for named entity recognition as well as training different word embedding models for the Amazigh language to be available for NLP projects.

#### REFERENCES




- [1] J. H. Greenberg, "The languages of Africa," *Bloomington Indiana Univ. The Hague Mouton*, 1966.
- [2] O. Ouakrim, "Fonética y fonología del Bereber. Survey: University of Autònoma de Barcelona," *University of Autònoma de Barcelona*, 1995.
- [3] S. Amri, L. Zenkouar, and M. Outahajala, "Build a morphosyntactically annotated amazigh corpus," in *ACM International Conference Proceeding Series*, Mar. 2017, vol. Part F1294, pp. 1–7, doi: 10.1145/3090354.3090362.
- [4] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 1966, doi: 10.1214/aoms/1177699147.
- [5] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data abstract," *Proceedings of the 18th International Conference on Machine Learning*, vol. 2001, no. June, pp. 282–289, 1999.
- [6] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/bf00994018.
- [7] M. Outahajala, Y. Benajiba, P. Rosso, and L. Zenkouar, "POS tagging in Amazighe using support vector machines and conditional random fields," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, vol. 6716 LNCS, pp. 238–241, doi: 10.1007/978-3-642-22327-3\_28.
- [8] A. Samir, Z. Lahbib, and O. Mohamed, "Amazigh PoS tagging using machine learning techniques," in *Lecture Notes in Networks and Systems*, 2018, vol. 37, pp. 551–562, doi: 10.1007/978-3-319-74500-8\_51.
- [9] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," *Proceedings of International Conference on New Methods in Language Processing*, pp. 44–49, 1994, [Online]. Available: <https://www.semanticscholar.org/paper/Probabilistic-part-of-speech-tagging-using-decision-Schmidt/bd0bab6fc8cd43c0ce170ad2f4cb34181b31277d>.
- [10] P. Denis and B. Sagot, "Coupling an annotated corpus and a lexicon for state-of-the-art PoS tagging," *Language Resources and Evaluation*, vol. 46, no. 4, pp. 721–736, Dec. 2012, doi: 10.1007/s10579-012-9193-0.
- [11] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5–6, pp. 602–610, Jul. 2005, doi: 10.1016/j.neunet.2005.06.042.
- [12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [13] P. Liu, S. Joty, and H. Meng, "Fine-grained opinion mining with recurrent neural networks and word embeddings," in *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1433–1443, doi: 10.18653/v1/d15-1168.
- [14] J. Zhou and W. Xu, "End-To-end learning of semantic role labeling using recurrent neural networks," in *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, 2015, vol. 1, pp. 1127–1137, doi: 10.3115/v1/p15-1109.
- [15] M. Ballesteros, C. Dyer, and N. A. Smith, "Improved transition-based parsing by modeling characters instead of words with LSTMs," in *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 349–359, doi: 10.18653/v1/d15-1041.
- [16] E. Kipervasser and Y. Goldberg, "Simple and accurate dependency parsing using bidirectional LSTM feature representations," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 313–327, Dec. 2016, doi: 10.1162/tac1\_a\_00101.
- [17] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, [Online]. Available: <http://arxiv.org/abs/1508.01991>.
- [18] W. Ling et al., "Finding function in form: Compositional character models for open vocabulary word representation," in *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1520–1530, doi: 10.18653/v1/d15-1176.
- [19] P. Wang, Y. Qian, F. K. Soong, L. He, and H. Zhao, "A unified tagging solution: Bidirectional LSTM recurrent neural network with word embedding," 2015, [Online]. Available: <http://arxiv.org/abs/1511.00215>.
- [20] B. Plank, A. Sogaard, and Y. Goldberg, "Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss," arXiv, Jul. 2016, [Online]. Available: <http://arxiv.org/abs/1604.05529>.
- [21] Y. Goldberg, "A primer on neural network models for natural language processing," *Journal of Artificial Intelligence Research*, vol. 57, pp. 345–420, Nov. 2016, doi: 10.1613/jair.4992.
- [22] K. Cho, "Natural language understanding with distributed representation," 2015, [Online]. Available: <http://arxiv.org/abs/1511.07916>.
- [23] J. F. Kolen and S. C. Kremer, "Gradient flow in recurrent nets: The difficulty of learning longterm dependencies," in *A Field Guide to Dynamical Recurrent Networks*, IEEE, 2010.
- [24] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 6645–6649, 2013, doi: 10.1109/ICASSP.2013.6638947.
- [25] A. Graves, *Supervised sequence labelling with recurrent neural networks*, vol. 385. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.
- [26] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, 2013.






**BIOGRAPHIES OF AUTHORS**

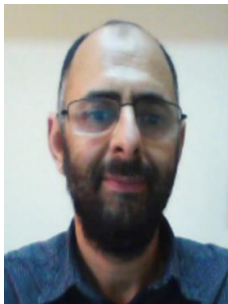
**Rkia Bani**    is a PhD candidate at the Mohammadia school of engineering (EMI), Mohammed V University in Rabat, Morocco. Rkia got a national telecom engineer diploma in 2011 from EMI engineering school. Rkia works as a project health manager. She can be contacted at email: [rkiabani@research.emi.ac.ma](mailto:rkiabani@research.emi.ac.ma).






**Samir Amri**    got his engineering degree from Mohammadia School of Engineering in 2006 and PhD degree from the same school. Samir worked as a senior consultant in information and communication technology and project management. Now, he is an assistant lecturer in ENSAM school of engineering in Meknes. His fields of interest include Artificial intelligence, communication and information technology applied to Amazigh language. He can be contacted at email: [amri.samir@gmail.com](mailto:amri.samir@gmail.com).



**Lahbib Zenkour**    Received the Dr. Eng. degree from CEM, Université des Sciences et Techniques du Languedoc, Montpellier, France in 1983 and PhD degree from ULg (Liège) in Belgium. After working as an assistant lecturer in the Mohammadia School of Engineering in Rabat, he has been a professor lecturer since 1996 in the same school. His fields of interest include signal processing, information technology, telecommunications, and communication and information technology applied to Amazigh language. He can be contacted at email: [lahbib.zenkour@gmail.com](mailto:lahbib.zenkour@gmail.com).



**Zouhair Guennoun**    Received his engineering degree from the Electronics and Electrical Montefiore Institute, ULG Liege, Belgium in 1987; his M.Sc. degree from the EMI School of Engineering, Rabat, Morocco in 1993; and his PhD degree from the same school in 1996. From 1988-1996 he worked as an assistant lecturer in the EMI School of Engineering, and from 1996 he has been working in the same school as a professor lecturer. His fields of interest are digital signal processing, error control coding, speech, and image processing. He can be contacted at email: [zouhair@emi.ac.ma](mailto:zouhair@emi.ac.ma).